



SystemC CCI WG Parameter Information

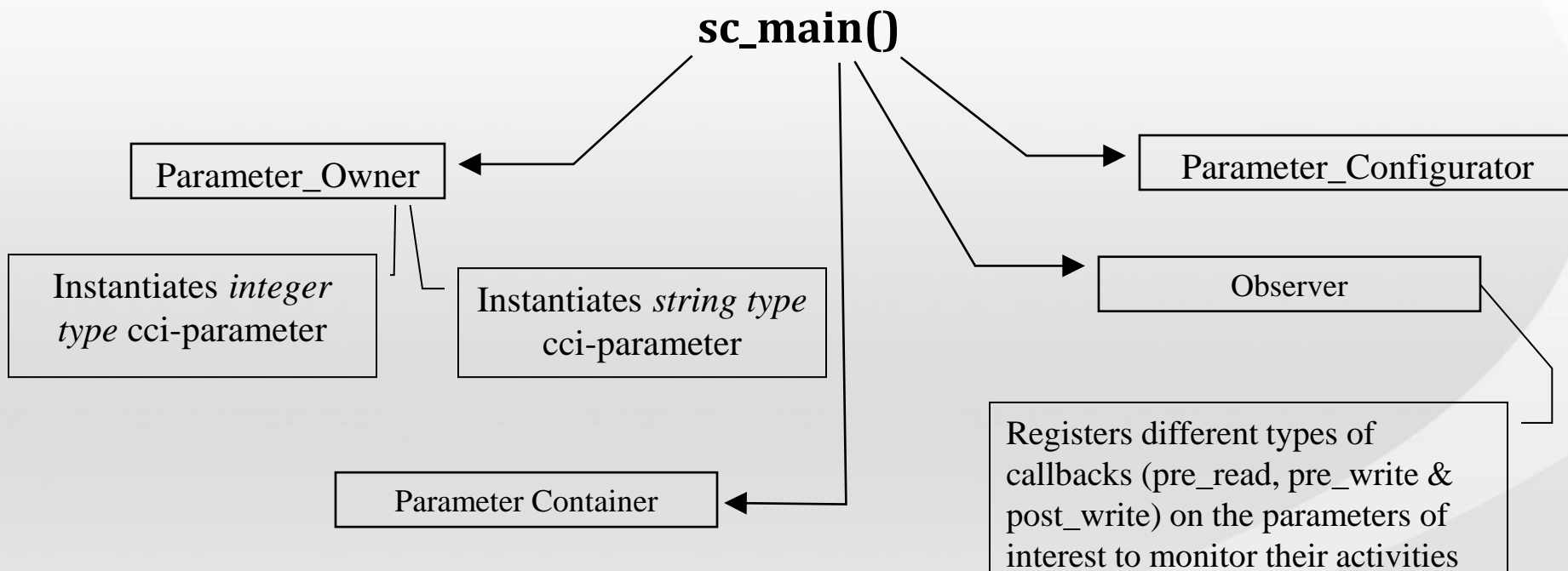
P V S Phaneendra, CircuitSutra Technologies Pvt. Ltd.
December 2011



Set and Query Various Attributes of a CCI Parameter

- The following functionality have been demonstrated in this example :
 - Setting and querying
 - Parameter's name
 - Parameter's value
 - Parameter's description/documentation/meta-data
 - Parameter's value validity
 - Parameter's locked status
 - Parameter's value origin

Set and Query Parameter Information



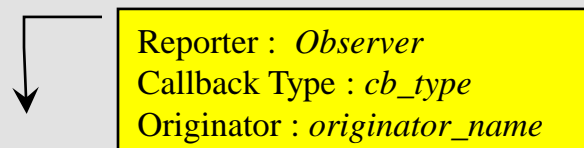
Note#1 : For a detailed understanding of registration and implementation of callbacks, please go see examples 8, 11 and 12.

Within the Callbacks implementation :

Name of the parameter : `callback_event.name();`

Name of the originator : `callback_event.originator.name();`

Note#2 : The originator information will be displayed by the following **Shape**



Set and Query Parameter Information (1)

sc_main()

Register a global broker

```
cci::cci_register_broker(new cci_utils::broker("DEFAULT_BROKER"));
```

Instantiate a cci_originator instance to get access to the *DEFAULT BROKER*

```
cci::cci_originator me("sc_main");
```

Get handle to the *DEFAULT BROKER* using the above *Originator*

```
cci::cci_broker_handle globalBroker =  
cci::cci_broker_manager::get_broker(me);
```

Set preset value using broker handle that overrides the default value passed as argument to the constructor of the sc_module (S)

```
globerBroker.set_preset_cci_value("param_owner.mutable_string_param",  
cci_value("Initialized within sc_main"));
```

Demonstrating value comparison for different mutability types with the same data type

All the three mutability type cci-parameters of std::string type inside parameter container:

```
cci::cci_param< std::string, cci::CCI_MUTABLE_PARAM > mutab_str_param  
("string_mutab_param","String_Value_A");
```

Set and Query Parameter Information (2)

sc_main()

```
cci::cci_param< std::string, cci::CCI_IMMUTABLE_PARAM> immutab_str_param  
("string_immutab_param", "String_Value_A");
```

Query (S) : Is "mutab_str_param.get_value() == immutable_str_param.get_value()" ?

Returns : TRUE

Prior to the instantiation of the modules

Parameter Owner

Parameter Configurer

Create instances of the modules now

```
parameter_owner    param_owner("param_owner");  
parameter_configurer param_cfgr("param_cfgr");  
observer           observer_class("observer");
```

(I) -> Integer
(S) -> String

Set and Query Parameter Information (3)

Parameter Owner

```
cci::cci_param<int> int_param;  
cci::cci_param<std::string> string_param;
```

Set name and assign default value **(I)** :

```
int_param("mutable_int_param, 0);
```

Set name and assign default value **(S)** :

```
string_param("mutable_string_param",  
"Default_value");
```

Query **(I)** : `int_param.is_preset_value()`
returns **FALSE**

Query default value **(I)** :

```
int_param.get_default_value();  
Returns Value = 0;
```

Set documentation **(I)** :

```
const std::string init_doc =  
"This is a mutable type integer parameter";  
int_param.set_documentation(init_doc);
```

Prior to 0ns

Parameter Configurer

Declare base pointers and check the existence of the integer and string type parameters:

```
cci::cci_param_handle int_param_handle; (I)  
cci::cci_param_handle str_param_handle; (S)
```

Query **(S)** : `str_param_ptr->is_preset_value()`
returns **TRUE**

(I) -> Integer
(S) -> String

Set and Query Parameter Information (4)

Parameter Owner

Parameter Configurer

@ 0ns

Query (I) : `int_param_handle.is_default_value()` returns **TRUE**

@ 1ns

Set value (I) : `int_param = 1;`

Reporter : *Observer*
Callback Type : *pre_write, post_write*
Originator : *param_owner*

@ 2ns

Query (I) : `int_param_handle.is_default_value()` returns **FALSE**

Query (S) : `str_param_handle.is_default_value()` returns **TRUE**

Query Name (S) : `str_param_handle.name();`
Returns name = "param_owner.mutable_string_param";

Query Value (S) : `str_param_ptr.get_cci_value().get_string();`
Returns value = "Initialized within sc_main"

Set and Query Parameter Information (5)

Parameter Owner

@ 4ns

Reporter : *Observer*
Callback Type : *pre_read*
Originator : *param_cfgr*

Parameter Configurer

Query Name (I) : `int_param_handle.name()`;
Returns Name = "param_owner.mutable_int_param"

Query Value (I) : `int_param_ptr.get_cci_value()`;
Returns Value = "1"

Query Description (I) : `int_param_ptr.get_description()`;
Returns doc = "This is mutable type integer parameter";

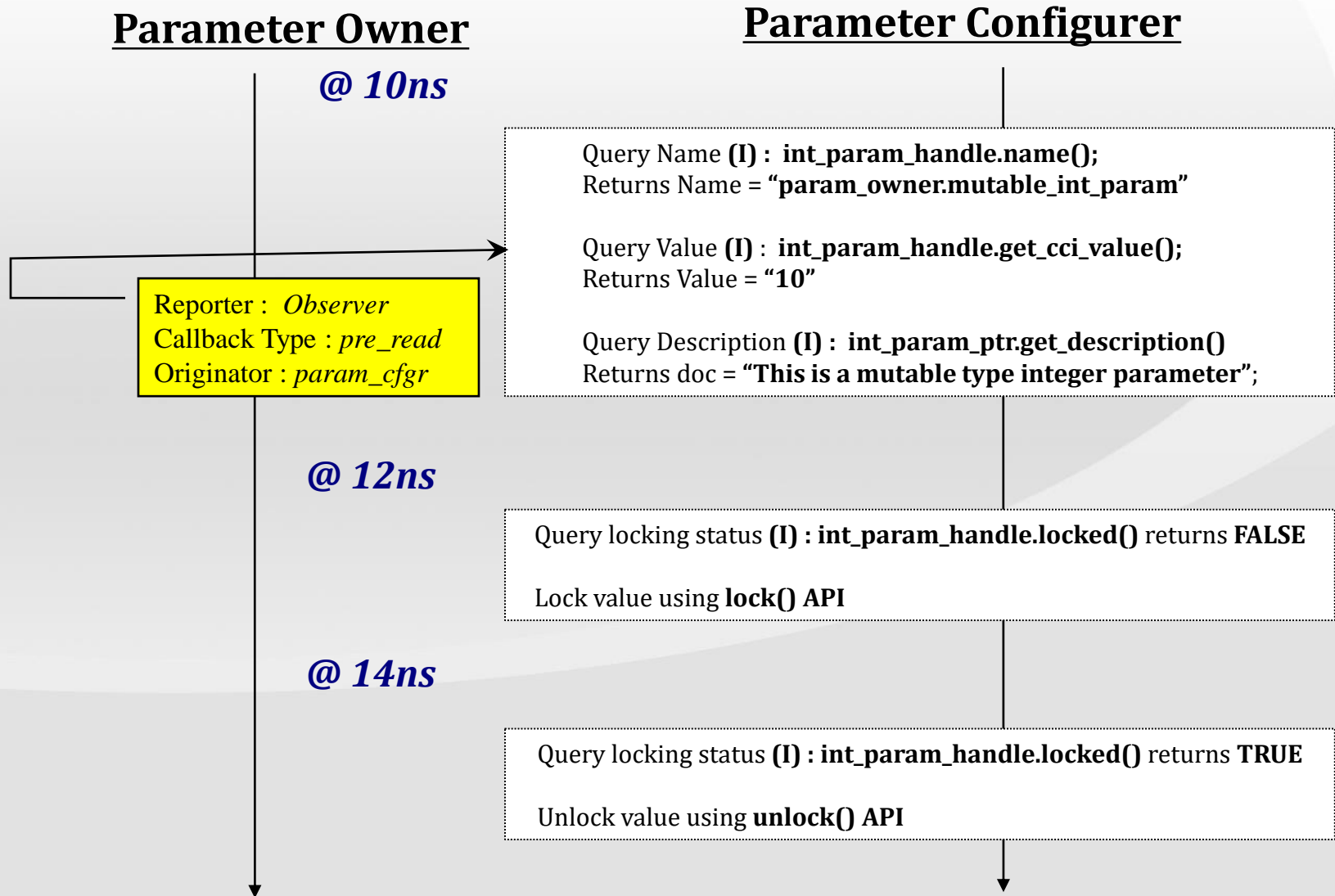
Query Metadata (I) : `int_param_ptr.get_metadata()`

@ 6ns

Set Value (I) :
`int_param_handle.set_cci_value(cci::cci_value(10));`

Reporter : *Observer*
Callback Type : *pre_write, post_write*
Originator : *param_cfgr*

Set and Query Parameter Information (6)



Set and Query Parameter Information (7)

Parameter Owner

Parameter Configurer

@ 16ns

Query locking status (I) : *int_param_handle.locked()* returns **FALSE**

@ 22ns

Query Documentation (S) : *str_param_handle.get_description()*
Returns doc = "This is a mutable type string parameter";

Query Parameter Basic Type (S) : *str_param_handle.get_basic_type()*;
Return Value = "4" (std::string type)

Set new value to string parameter using CCI_VALUE
cci::cci_value str_value("Hello_New_String");
str_param_handle.set_cci_value(str_value);

Set and Query Parameter Information (8)

Parameter Owner

@ 24ns

Parameter Configurer

Query the value origin (S) :

*const cci::cci_originator**

str_originator = str_param_handle.get_value_origin();

Name of the value origin: *str_originator.name();*

Returns Originator : *param_cfgr*

Query Value (S) : *str_param_handle.get_cci_value().to_json();*

Returns Value = "Hello_New_String"

Create an instance of cci_value to receive value into cci_value object

cci::cci_value rx_value;

Query CCI Value (S) : *rx_value = str_param_ptr.get_cci_value();*

Query String Value (S) : *rx_value.get_string();*

Returned String : "Hello_New_String"

Set and Query Parameter Information (9)

Parameter Owner

Parameter Configurer

@ 26ns

Set new value to string parameter using CCI_VALUE

```
cci::cci_value str_value("String_Value_Set_by_OWNER");  
string_param.set_cci_value(str_value);
```

@ 28ns

Create an object of **cci_value** to receive value into cci_value object

```
cci::cci_value rec_str_value;
```

Query CCI Value (S) :

```
rec_str_value = string_param.get_cci_value();
```

Query (S) : new string value using CCI_VALUE :

```
std::string recv_str = rec_str_value.get_string();  
Returns string : "String_Value_Set_by_OWNER"
```

Expected Output

(ex07_Parameter_Information.log)

SystemC Simulation

Info: sc_main: [MAIN] : Setting 'param_owner.mutable_string_param' value to 'Initialized within sc_main()'

Info: sc_main: [MAIN] : Demonstrating 'comparison' between the values of a data type for different mutability types

Info: sc_main: [MAIN] : 'mutable' & 'immutable' type String parameters - VALUES MATCH

Info: sc_main: [MAIN] : parameter_owner module declares two cci type parameters.

Info: sc_main: One is of 'Integer type' and the other is of 'String type'

Info: sc_main: 'Integer type' has 'pre_read', 'pre/post_write' callback registered within the OBSERVER

Info: param_owner: @0 s, Prior to 0 s demonstrating 'is_preset_value' API

Info: param_owner: @0 s, [OWNER] : Parameter : param_owner.mutable_int_paramValue : 0

Info: param_owner: @0 s, Is preset value ? false

Info: param_owner: @0 s, Prior to 0 s demonstrating 'get_default_value()'

Info: param_owner: @0 s, [OWNER -> Retrieve] : Parameter name : param_owner.mutable_int_param

Info: param_owner: @0 s, [OWNER -> Retrieve] : Using 'get_default_value()' : 0

Cont'd

Info: param_owner: @0 s, [OWNER -> Set] : Param description - 'This is a mutable type integer parameter'

Info: param_cfgr: @0 s, [CFGR C_TOR] : Broker Type : DEFAULT_BROKER - is not a private broker.

Info: param_cfgr: @0 s, Prior to 0 s demonstrating 'is_preset_value' API

Info: param_cfgr: @0 s, [CFGR C_TOR] : param_owner.mutable_string_param Preset Value :
Initialized within sc_main

Info: param_cfgr: @0 s, [CFGR C_TOR] : Is preset value ? true

Info: observer: @0 s, [OBSERVER C_TOR] : Broker Type : DEFAULT_BROKER- is not a private broker.

Info: sc_main: Begin Simulation

Info: param_cfgr: @0 s, @ 0 s demonstrating 'is_default_value()'

Info: param_cfgr: @0 s, [CFGR] : param_owner.mutable_int_param default value hasn't been modified.

Info: param_cfgr: @0 s, [CFGR] : Is Default Value ? Returned status : true

Info: param_owner: @1 ns, @ 1 ns

Info: param_owner: @1 ns, [OWNER -> Set] : Overriding default value of param_owner.mutable_int_param to 1

Info: observer: @1 ns, [OBSERVER pre_write_cb] : Parameter Name : param_owner.mutable_int_param
Originator info : param_owner

Info: observer: @1 ns, [OBSERVER post_write_cb] : Parameter Name : param_owner.mutable_int_param
Originator info : param_owner

Info: param_cfgr: @2 ns, @ 2 ns demonstrating 'is_default_value()'

Cont'd

Info: param_cfgr: @2 ns, [CFGR] : param_owner.mutable_int_param value has been modified.

Info: param_cfgr: @2 ns, [CFGR] : Is Default Value ? Returned status : false

Info: param_cfgr: @2 ns, [CFGR] : param_owner.mutable_string_param New Value : Initialized within sc_main

Info: param_cfgr: @4 ns, [CFGR -> Retrieve] Parameter's name : param_owner.mutable_int_param

Info: observer: @4 ns, [OBSERVER pre_read_cb] : Parameter Name : param_owner.mutable_int_param
Originator info : param_cfgr

Info: param_cfgr: @4 ns, [CFGR -> Retrieve] Parameter's value : 1

Info: param_cfgr: @4 ns, [CFGR -> Retrieve] Parameter's desc : This is a mutable type integer parameter

Info: param_cfgr: @4 ns, [CFGR -> Retrieve] Parameter's metadata : {"min_value":["0","Minimum value"],"max_value":["100","Maximum value"],"unit":["F","Unit of the parameter (Farad)"]}

Info: param_cfgr: @6 ns, @ 6 ns

Info: param_cfgr: @6 ns, [CFGR -> Set] : param_owner.mutable_int_param value to 10

Info: observer: @6 ns, [OBSERVER pre_write_cb] : Parameter Name : param_owner.mutable_int_param
Originator info : param_cfgr

Info: observer: @6 ns, [OBSERVER post_write_cb] : Parameter Name : param_owner.mutable_int_param
Originator info : param_cfgr

Info: param_cfgr: @10 ns, @ 10 ns

Cont'd

```
Info: param_cfgr: @10 ns, [CFGR -> Retrieve] : Parameter name : param_owner.mutable_int_param
Info: observer: @10 ns, [OBSERVER pre_read_cb] : Parameter Name : param_owner.mutable_int_param
Originator info : param_cfgr
Info: param_cfgr: @10 ns, [CFGR -> Retrieve] : Parameter value: 10
Info: param_cfgr: @10 ns, [CFGR -> Retrieve] : Parameter desc : This is a mutable type integer
parameter
Info: param_cfgr: @12 ns, @ 12 ns
Info: param_cfgr: @12 ns, [CFGR] : Checking locked status of parameter
Info: param_cfgr: @12 ns, [CFGR] : param_owner.mutable_int_param is in unlocked state!!
Info: param_cfgr: @12 ns, [CFGR] : Parameter locked using password
Info: param_cfgr: @14 ns, @ 14 ns
Info: param_cfgr: @14 ns, [CFGR] : Checking locked status of parameter
Info: param_cfgr: @14 ns, [CFGR] : param_owner.mutable_int_param is in locked state!!
Info: param_cfgr: @14 ns, [CFGR] : Parameter unlocked using password
Info: param_cfgr: @16 ns, @ 16 ns
Info: param_cfgr: @16 ns, [CFGR] : Checking locked status of parameter
Info: param_cfgr: @16 ns, [CFGR] : param_owner.mutable_int_param is in unlocked state!!
Info: param_cfgr: @22 ns, @ 22 ns      demonstrating 'set_value()' for string-type param using
cci_value
```


Cont'd

```
Info: param_cfgr: @22 ns, [CFGR] : Parameter Description : This is a mutable type string parameter
Info: param_cfgr: @22 ns, [CFGR -> Set] : Get Basic Type using 'get_basic_type()' : 4
Info: param_cfgr: @24 ns, @ 24 ns demonstrating 'get_value_origin' API
Info: param_cfgr: @24 ns, [CFGR] : Originator for the latest write on string type cci-parameter :
param_cfgr
Info: param_cfgr: @24 ns, @ 24 ns demonstrating 'get_value' for string-type param using cci_value
Info: param_cfgr: @24 ns, [CFGR] : 'Retrieve1' using 'json_serialize' and 'Retrieve2' using
'get_value()'
Info: param_cfgr: @24 ns, [CFGR -> Retrieve1] : param_owner.mutable_string_param value is
"Hello_New_String"
Info: param_cfgr: @24 ns, [CFGR -> Retrieve2] : Hello_New_String
Info: param_owner: @26 ns, @ 26 ns      demonstrating setting values by OWNER using cci_value
Info: param_owner: @26 ns, [OWNER -> Set] : New String Value : 'String_Value_Set_by_OWNER
Info: param_owner: @28 ns, @ 28 ns      demonstrating retrieving values by OWNER using cci_value
Info: param_owner: @28 ns, [OWNER -> Retrieve] : Receive str_value using 'cci_value' :
String_Value_Set_by_OWNER
Info: sc_main: End Simulation
```